



# UNITED STATES PATENT AND TRADEMARK OFFICE

A

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/506,125	02/17/2000	Felix G. T. I. Andrew	202411	6776

7590 07/26/2005

Leydig Voit & Mayer Ltd  
Two Prudential Plaza  
1800 North Stetson Suite 4900  
Chicago, IL 60601-6780

EXAMINER
----------

KISS, ERIC B

ART UNIT	PAPER NUMBER
----------	--------------

2192

DATE MAILED: 07/26/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

# Office Action Summary

Application No.

09/506,125

Applicant(s)

ANDREW ET AL.

Examiner

Eric B. Kiss

Art Unit

2192

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

## Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

## Status

- 1) ☒ Responsive to communication(s) filed on 26 April 2005.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

## Disposition of Claims

- 4) ☒ Claim(s) 26-53 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 26-53 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

## Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

## Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
  - ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

## Attachment(s)

- |   |   |
|---|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892)  | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date: _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)  | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152)             |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)<br>Paper No(s)/Mail Date <u>20050324</u> . | 6) <input type="checkbox"/> Other: _____  |

PD

## **DETAILED ACTION**

### ***Continued Examination Under 37 CFR 1.114***

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 26 April 2005 has been entered.

Claims 26-53 are pending.

### ***Response to Arguments***

2. Applicant's arguments with respect to claims 26-53 have been considered but are moot in view of the new ground(s) of rejection.

### ***Admitted Prior Art***

3. If Applicant does not seasonably traverse the well-known statement during examination, then the object of the well-known statement is taken to be admitted prior art. In re Chevenard, 139 F.2d 71, 60 USPQ 239 (CCPA 1943). A seasonable challenge constitutes a demand for evidence made as soon as practicable during prosecution. Thus, Applicant is charged with rebutting the well-known statement in the next reply after the Office action in which the well-known statement was made. This is necessary because the Examiner must be given the opportunity to provide evidence in the next Office action or explain why no evidence is required. If the Examiner adds a reference to the rejection in the next action after applicant's rebuttal, the

newly cited reference, if it is added merely as evidence of the prior well known statement, does not result in a new issue and thus the action can potentially be made final.

4. The following statement is taken to be admitted prior art:

It has been well known and practiced in the software development art for multiple developers in a collaborative development environment to use multiple computers connected by way of a network. (See the unchallenged statement of Official Notice taken in the rejections of claim 26 in the Office actions mailed 5 November 2004 and 6 May 2004).

***Claim Rejections - 35 USC § 103***

5. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

6. Claims 34-37 and 42-53 are rejected under 35 U.S.C. 103(a) as being unpatentable over the Adobe® FrameMaker®+SGML integrated XML authoring and composition tool as disclosed by “Adobe® FrameMaker®+SGML 5.5: Developing SGML Publishing Applications,” 1997 (hereinafter *AFM97*) and Charles F. Goldfarb and Paul Prescod, “The XML Handbook,” 1998 (hereinafter *Goldfarb and Prescod*), pp. 278-295, in view of Elliotte Rusty Harold, “XML Bible,” 1999, IDG Books Worldwide, Inc. (art of record; hereinafter *Harold*).

As per claims 34 and 37, *Goldfarb and Prescod* disclose a set of one or more routines for modifying at least one user interface resource file (see, for example, section 21.2 and its corresponding subsections on pages 280-288); and the at least one resource file comprising a document in a markup language, wherein tagged text elements are associated with attributes of a

Art Unit: 2192

user interface (Adobe® FrameMaker® is disclosed as a tool for creating and editing XML documents; see, for example, section 21.2 and its corresponding subsections on pages 280-288). *Goldfarb and Prescod* disclose software for modifying the at least one user interface file ((see, for example, section 21.3 and its corresponding subsections on pages 288-290; the product has a WYSIWYG environment that enables interactive editing of a document, allowing the user to see the rendered result as changes are made to the formatting specification. Changing the rendered (WYSIWYG) display and/or the structured view of the document is, in effect, changing the portion of the FrameMaker®+SGML tool with which the user interacts (namely, the manipulatable graphical display), and thus user interface of the FrameMaker®+SGML tool is changed). *Goldfarb and Prescod* further disclose a computer-readable medium storing computer-executable instructions and computer-readable data for implementing the aforementioned components (see, for example, page 278, indicating a free trial version of FrameMaker®+SGML on CD-ROM). The use of a computer including memory for storing the executable program is inherent in performing the aforementioned computer-implemented steps. The FrameMaker®+SGML tool is not expressly disclosed with the markup language including a plurality of resource tag attribute default value mechanisms whose order corresponds to a precedence order of default values for the attributes. However, *Harold* teaches such a plurality of markup language mechanisms (see, for example, the discussion of Inheritance on pp. 334-335). Therefore, it would have been obvious to one of ordinary skill in the computer art at the time the invention was made to modify the system and medium of the FrameMaker®+SGML tool to include such mechanisms in order to simplify the specification of formatting rules.

As per claim 35, *Goldfarb and Prescod* further disclose the routines for modifying the at least one user interface resource file being invoked while the computer program is being executed, the customizing occurring dynamically (see, for example, section 21.3 and its corresponding subsections on pages 288-290; the product has a WYSIWYG environment that enables interactive editing of a document, allowing the user to see the rendered result as changes are made to the formatting specification. Changing the rendered (WYSIWYG) display and/or the structured view of the document is, in effect, changing the portion of the FrameMaker®+SGML tool with which the user interacts (namely, the manipulatable graphical display), and thus user interface of the FrameMaker®+SGML tool is changed).

As per claim 36, *Goldfarb and Prescod* further disclose a set of operating system resource-loading routines for presenting the user interface to the user, wherein the resource-loading routines obtain user interface resource information from a user interface attribute data tree corresponding to the user interface resource file and, with respect to resource information not specified in the user interface resource file, from a set of default sources of user interface resource information (see, for example, section 21.2.1 on page 281; and section 21.2.5 on page 288).

As per claim 42, *Goldfarb and Prescod* disclose the routines for creating and modifying the at least one user interface resource file being used while the computer program is being executed, the creating and modifying occurring dynamically and not requiring a recompilation of the executable program component (see, for example, section 21.3 and its corresponding subsections on pages 288-290; the product has a WYSIWYG environment that enables interactive editing of a document, allowing the user to see the rendered result as changes are

made to the formatting specification. Changing the rendered (WYSIWYG) display and/or the structured view of the document is, in effect, changing the portion of the FrameMaker®+SGML tool with which the user interacts (namely, the manipulatable graphical display), and thus user interface of the FrameMaker®+SGML tool is changed). The FrameMaker®+SGML tool is not expressly disclosed with the markup language including a plurality of resource tag attribute default value mechanisms whose order corresponds to a precedence order of default values for the attributes. However, *Harold* teaches such a plurality of markup language mechanisms (see, for example, the discussion of Inheritance on pp. 334-335). Therefore, it would have been obvious to one of ordinary skill in the computer art at the time the invention was made to modify the medium of the FrameMaker®+SGML tool to include such mechanisms in order to simplify the specification of formatting rules.

As per claims 43 and 46, *Goldfarb and Prescod* disclose executing a computer program, thereby causing a user interface to be presented (see, for example, section 21.3 and its corresponding subsections on pages 288-290); editing at least one user interface resource file, the at least one user interface file comprising a document in a markup language, wherein tagged text elements are associated with attributes of the user interface (Adobe® FrameMaker® is disclosed as a tool for creating and editing XML documents; see, for example, section 21.2 and its corresponding subsections on pages 280-288); and causing a new user interface to be presented (see, for example, section 21.3 and its corresponding subsections on pages 288-290; the product has a WYSIWYG environment that enables interactive editing of a document, allowing the user to see the rendered result as changes are made to the formatting specification. Changing the rendered (WYSIWYG) display and/or the structured view of the document is, in effect, changing

the portion of the FrameMaker®+SGML tool with which the user interacts (namely, the manipulatable graphical display), and thus user interface of the FrameMaker®+SGML tool is changed). *Goldfarb and Prescod* further discloses a computer-readable medium storing computer-executable instructions and computer-readable data for implementing the aforementioned steps (see, for example, page 278, indicating a free trial version of FrameMaker®+SGML on CD-ROM). The FrameMaker®+SGML tool is not expressly disclosed with the markup language including a plurality of resource tag attribute default value mechanisms whose order corresponds to a precedence order of default values for the attributes. However, *Harold* teaches such a plurality of markup language mechanisms (see, for example, the discussion of Inheritance on pp. 334-335). Therefore, it would have been obvious to one of ordinary skill in the computer art at the time the invention was made to modify the system and medium of the FrameMaker®+SGML tool to include such mechanisms in order to simplify the specification of formatting rules.

As per claim 44, *Goldfarb and Prescod* further discloses parsing the at least one user interface resource file into a user interface attribute data tree (see, for example, section 21.2.1 on page 281); invoking operating system resource-loading routines for constructing the user interface (see, for example, section 21.3 and its corresponding subsections on pages 288-290; the product has a WYSIWYG environment that enables interactive editing of a document, allowing the user to see the rendered result as changes are made to the formatting specification. Changing the rendered (WYSIWYG) display and/or the structured view of the document is, in effect, changing the portion of the FrameMaker®+SGML tool with which the user interacts (namely,



the manipulatable graphical display), and thus user interface of the FrameMaker®+SGML tool is changed); and obtaining user interface resource information from the user interface attribute data tree, and with respect to resource information not specified in the user interface resource file, from a set of default sources of user interface resource information (see, for example, section 21.2.1 on page 281; and section 21.2.5 on page 288).

As per claim 45, *Goldfarb and Prescod* further discloses causing the user interface to be presented occurring while the computer program is being executed and not requiring the computer program to be re-executed (see, for example, section 21.3 and its corresponding subsections on pages 288-290; the product has a WYSIWYG environment that enables interactive editing of a document, allowing the user to see the rendered result as changes are made to the formatting specification. Changing the rendered (WYSIWYG) display and/or the structured view of the document is, in effect, changing the portion of the FrameMaker®+SGML tool with which the user interacts (namely, the manipulatable graphical display), and thus user interface of the FrameMaker®+SGML tool is changed).

As per claims 47-53, see the relevant teachings of *Harold* applied above, where it is further taught that a child node may inherit attribute data default values from a parent node, with any conflicting attribute values in the child node taking precedence over otherwise inherited parent attributes (see, for example, the discussion of Inheritance on pp. 334-335). For reasons stated above, such claims also would have been obvious.

7. Claims 26-33 and 38-41 are rejected under 35 U.S.C. 103(a) as being unpatentable over the Adobe® FrameMaker®+SGML integrated XML authoring and composition tool as disclosed

Art Unit: 2192

by “Adobe® FrameMaker®+SGML 5.5: Developing SGML Publishing Applications,” 1997 (hereinafter *AFM97*) and Charles F. Goldfarb and Paul Prescod, “The XML Handbook,” 1998 (hereinafter *Goldfarb and Prescod*), pp. 278-295, in view of *Harold*.

As per claim 26, *Goldfarb and Prescod* disclose a set of one or more routines for producing an executable program component of the computer application software product (see, for example, section 21.2 and its corresponding subsections on pages 280-288); at least one resource file comprising a document in a markup language, wherein tagged text elements are associated with attributes of a user interface (Adobe® FrameMaker® is disclosed as a tool for creating and editing XML documents; see, for example, section 21.2 and its corresponding subsections on pages 280-288 of *Goldfarb and Prescod*); and a set of one or more routines for creating and modifying the user interface component by manipulating the at least one user interface resource file (see, for example, section 21.2 and its corresponding subsections on pages 280-288). *AFM97* discloses, as part of application development with the FrameMaker®+SGML tool, various skills required within the implementation team, including document design, SGML knowledge, setting up FrameMaker®\_SGML formatting templates, and setting up the formatting rules that control automatic application of the desired graphic design to structure documents. Further *AFM97* discloses these skills as being shared by the group of people rather than being mastered by one individual (see, for example, “The implementation team” on pp. 7-8). FrameMaker®+SGML is not expressly disclosed as being used in a development environment comprising at a first computer and a second computer linked by way of a network. However, as admitted prior art, it has been well known and practiced in the software development art for

multiple developers in a collaborative development environment (as disclosed, for example, by *AFM97* on pp. 7-8) to use multiple computers connected by way of a network. Therefore, it would have been obvious to one having ordinary skill in the computer art at the time the invention was made to use the FrameMaker®+SGML tool in a development environment comprising at a first computer and a second computer linked by way of a network. One would be motivated to do so to promote more cost effective and efficient software development. The FrameMaker®+SGML tool is not expressly disclosed with the markup language including a plurality of resource tag attribute default value mechanisms whose order corresponds to a precedence order of default values for the attributes. However, *Harold* teaches such a plurality of markup language mechanisms (see, for example, the discussion of Inheritance on pp. 334-335). Therefore, it would have been obvious to one of ordinary skill in the computer art at the time the invention was made to modify the system of the FrameMaker®+SGML tool to include such mechanisms in order to simplify the specification of formatting rules.

As per claim 27, *Goldfarb and Prescod* further disclose the routines for creating and modifying the at least one user interface resource file being used while the computer program is being executed, the creating and modifying occurring dynamically and not requiring a recompilation of the executable program component (see, for example, section 21.3 and its corresponding subsections on pages 288-290; the product has a WYSIWYG environment that enables interactive editing of a document, allowing the user to see the rendered result as changes are made to the formatting specification. Changing the rendered (WYSIWYG) display and/or the structured view of the document is, in effect, changing the portion of the FrameMaker®+SGML tool with which the user interacts (namely, the manipulatable graphical display), and thus user

interface of the FrameMaker®+SGML tool is changed). Therefore, for reasons applied above, such a claim also would have been obvious.

As per claims 28 and 30-32, *Goldfarb and Prescod* further disclose a set of operating system resource-loading routines for presenting a user interface corresponding to the user interface component (see, for example, see, for example, section 21.2.1 on page 281). Further, as the FrameMaker®+SGML tool is a collaborative tool (see, for example, *Goldfarb and Prescod*, section 21.4; and *AFM97*, “The implementation team” on pages 7-8), the generated user interface can be viewed by any member of the implementation team. Therefore, for reasons applied above, such claims also would have been obvious.

As per claim 29, *Goldfarb and Prescod* further disclose the resource-loading routines obtain user interface resource information from a user interface attribute data tree corresponding to the user interface resource file and, with respect to resource information not specified in the user interface resource file, from a set of default sources of user interface resource information (see, for example, section 21.2.1 on page 281; and section 21.2.5 on page 288). Therefore, for reasons applied above, such a claim also would have been obvious.

As per claim 33, *Goldfarb and Prescod* further disclose a computer-readable medium storing computer-executable instructions and computer-readable data for implementing the aforementioned components (see, for example, page 278, indicating a free trial version of FrameMake®r+SGML on CD-ROM). Therefore, for reasons applied above, such a claim also would have been obvious.

As per claim 38, *Goldfarb and Prescod* disclose causing a user interface to be presented while a computer program is being executed and not requiring the computer program to be re-executed (see, for example, section 21.3 and its corresponding subsections on pages 288-290; the product has a WYSIWYG environment). In the WYSIWYG environment of FrameMaker®+SGML, making changes to the user interface can be done without necessarily requiring a change in the executable code, such as disclosed in *Goldfarb and Prescod*, section 21.3.2. Further, the FrameMaker®+SGML tool is a collaborative tool (see, for example, *Goldfarb and Prescod*, section 21.4; and *AFM97*, “The implementation team” on pages 7-8). *AFM97* discloses, as part of application development with the FrameMaker®+SGML tool, various skills required within the implementation team, including document design, SGML knowledge, setting up FrameMaker®\_SGML formatting templates, and setting up the formatting rules that control automatic application of the desired graphic design to structure documents. Further *AFM97* discloses these skills as being shared by the group of people rather than being mastered by one individual (see, for example, “The implementation team” on pp. 7-8). FrameMaker®+SGML is not expressly disclosed as being used in a development environment comprising at least two users -- one developer and one graphic designer, where the graphic designer proposes necessary functional changes to the developer. However, it is well known and practiced in the software development art to split up the tasks of an overall software development project into subtasks relating to programming and graphical interface design. This common practice is also acknowledged by Applicant on page 3, lines 9-11 of the specification. Therefore, it would have been obvious to one having ordinary skill in the computer art at the time the invention was made to use the FrameMaker®+SGML tool in a development environment

comprising at least two users -- one developer and one graphic designer, where the graphic designer proposes necessary functional changes to the developer. One would be motivated to do so to promote more cost effective and efficient software development. The FrameMaker®+SGML tool is not expressly disclosed with the markup language including a plurality of resource tag attribute default value mechanisms whose order corresponds to a precedence order of default values for the attributes. However, *Harold* teaches such a plurality of markup language mechanisms (see, for example, the discussion of Inheritance on pp. 334-335). Therefore, it would have been obvious to one of ordinary skill in the computer art at the time the invention was made to modify the method of the FrameMaker®+SGML tool to include such mechanisms in order to simplify the specification of formatting rules.

As per claim 39, *Goldfarb and Prescod* further disclose at least one resource file comprising a document in a markup language, wherein tagged text elements are associated with attributes of a user interface Adobe® FrameMaker® is disclosed as a tool for creating and editing XML documents; see, for example, section 21.2 and its corresponding subsections on pages 280-288 of *Goldfarb and Prescod*). Therefore, for reasons applied above, such a claim also would have been obvious.

As per claim 40, *Goldfarb and Prescod* further disclose parsing the at least one user interface resource file into a user interface attribute data tree (see, for example, section 21.2.1 on page 281); and obtaining user interface resource information from the user interface attribute data tree, and with respect to resource information not specified in the user interface resource file, from a set of default sources of user interface resource information (see, for example,

section 21.2.1 on page 281; and section 21.2.5 on page 288). Therefore, for reasons applied above, such a claim also would have been obvious.

As per claim 41, *Goldfarb and Prescod* further discloses causing the user interface to be presented occurring while the computer program is being executed and not requiring the computer program to be re-executed (see, for example, section 21.3 and its corresponding subsections on pages 288-290; the product has a WYSIWYG environment). Therefore, for reasons applied above, such a claim also would have been obvious.

### ***Conclusion***

8. Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Eric B. Kiss whose telephone number is (571) 272-3699. The Examiner can normally be reached on Tue. - Fri., 7:00 am - 4:30 pm. The Examiner can also be reached on alternate Mondays.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Tuan Dam, can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.


Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Art Unit: 2192

Any inquiry of a general nature should be directed to the TC 2100 Group receptionist:

571-272-2100.

EBK ~~EBK~~  
July 20, 2005



TUAN DAM  
SUPERVISORY PATENT EXAMINER